(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 2 377 063 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
**13.07.2016 Bulletin 2016/28**

(21) Application number: **09752257.7**

(22) Date of filing: **04.11.2009**

(51) Int Cl.:
***G06F 21/62*** *(2013.01)* ***G06F 3/06*** *(2006.01)*

(86) International application number:
**PCT/US2009/063260**

(87) International publication number:
**WO 2010/074817 (01.07.2010 Gazette 2010/26)**

(54) **METHOD AND APPARATUS FOR PROVIDING ACCESS TO FILES BASED ON USER IDENTITY**

VERFAHREN UND VORRICHTUNG ZUM BEREITSTELLEN VON ZUGANG ZU DATEIEN AUF DER BASIS VON BENUTZERIDENTITÄT

PROCÉDÉ ET APPAREIL PERMETTANT DE FOURNIR UN ACCÈS À DES FICHIERS SUR LA BASE DE L'IDENTITÉ DE L'UTILISATEUR

(84) Designated Contracting States:
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO SE SI SK SM TR**

(30) Priority: **26.12.2008 US 344407**

(43) Date of publication of application:
**19.10.2011 Bulletin 2011/42**

(73) Proprietor: **SanDisk IL Ltd.**
**44425 Kfar Saba (IL)**

(72) Inventor: **HAHN, Judah Gamliel**
**Ofra (IL)**

(74) Representative: **Prock, Thomas**
**Marks & Clerk LLP**
**90 Long Acre**
**London WC2E 9RA (GB)**

(56) References cited:
**EP-A1- 1 998 270**
**WO-A1-2007/099012**
**US-A1- 2005 033 721**
**US-A1- 2007 027 873**
**US-A1- 2007 283 094**
**US-A1- 2008 005 531**
**US-A1- 2008 098 023**



EP 2 377 063 B1

Printed by Jouve, 75001 PARIS (FR)

## Description

BACKGROUND

**[0001]** A device owner of a storage device has full rights to the contents therein for reading and writing. Often, the device owner wants to grant or deny access rights to other users and/or groups of individuals. File systems, such as the New Technology File System (NTFS) of Windows and some of the Linux file systems, permit the device owner to control access to files by assigning permissions for files.

**[0002]** By using file system technology, the device owner may safeguard files in both internal system storage and in external, portable storage such as USB flash drives (UFDs). However, this technology for safeguarding files may be circumvented in a portable storage device by simply connecting the portable storage device to a host that does not respect the permission rules of the file system. For example, the Linux NTFS driver ignores permission rules. Also, some third party drivers, such as the open-source ext2ifs that allow Windows to access ext2 partitions, do not enforce Linux permission rules. (The term "ext2" stands for "second extended file system," and the term "ext2ifs" references an installable file system (IFS) driver written by Stephan Schreiber, which is a driver implemented for some versions of the Microsoft Windows operating system.) Even connecting the portable storage device to a host running a different Windows domain will allow a local administrator to act as a device owner and consequently to override security measures.

**[0003]** Microsoft developed the Encrypting File System (EFS), which transparently encrypts the data within an NTFS and stores the keys within an Active Directory schema. Such approach ensures that if permissions are circumvented, the data within the files remains inaccessible. However, while this works well when integrated into Windows, EFS does not work in an independent environment and requires a central authentication mechanism in order to retrieve the keys.

**[0004]** The encryption technique Pretty Good Privacy (PGP) utilizes a transparent file encryption engine that uses shared key-rings. This technique provides a flexible software-based framework for maintenance and enforcement of permissions. However, this permission system is independent of the native operating system user/group permission set and requires additional management and client software, which must be installed on the host to access the data.

**[0005]** Hence, there exists a need for a way to safeguard files from unauthorized access, which works in an independent environment, does not require a central authentication mechanism to retrieve keys, and does not require additional management and client software installed on a host.

**[0006]** US 2007/0283094 discloses a portable storage system for connecting to a host, the portable storage system includes a storage device for storing information and a switch. The switch includes a get mode wherein the host sees only the free space in the storage device and not the part storing the information. Optionally, the portable storage system includes a give mode wherein the storage medium shows an empty space to the host and any file or directory is marked as shared and wherein the host sees a file-system whose size equals the amount of empty storage space on the storage device and an owner mode showing all of the stored information to the host and enabling the owner of the system to uncheck a shared flag on a storage device that received from another user that added files.

SUMMARY

**[0007]** According to an aspect of the present invention there is provided a method of providing a file system. The method comprises in a data storage device operatively coupled to a host device, performing commencing authentication of a user to the data storage device. If the authentication does not succeed, a third file system is provided to the host device coupled to the data storage device. Contents of the third file system are restricted to files authorized by public access rights. If the authentication does succeed it is determined whether the user is authenticated as having full rights to the data stored at the storage device by ascertaining ownership of a root directory of a first file system. In response to the user being authenticated as having full rights, the first file system is provided to the host device. The first file system is a native file system of the data storage device. The third file system includes a subset of the contents of the first file system. In response to the user not being authenticated as having full rights, a second file system is provided to the host device. The second file system is restricted to files that the user is authenticated as being authorized to access.

**[0008]** According to another aspect of the present invention there is provided a storage device. The storage device comprises a first memory module operative to store a first file system. The first file system is a native file system of the storage device. The device further comprises a second memory module operative to store generated data, an authentication module operative to authenticate a user and a controller operative to activate the authentication module and, in response to the user not being authenticated as having access rights other than public access rights, provide to the host device coupled to the data storage device a third file system. Contents of the third file system are restricted to files authorized by public access rights. The controller is further operative to, in response to the user being authenticated determine whether the user is authenticated as having full rights to the data stored at the storage device by ascertaining ownership of a root directory of a first file system. The controller is operative to, in response to the user being authenticated provide to the host device the first file system. The first file system is a native file system

of the data storage device. The third file system includes a subset of the contents of the first file system. The controller is further operative to, in response to the user not being authenticated provide to the host device a second file system, the second file system being restricted to files that the user is authenticated as being authorized to access.

**[0009]** In view of the foregoing, exemplary embodiments of the invention as described herein are designed for safeguarding files from unauthorized access and modification. However, the principles described may be implemented in alternative embodiments.

**[0010]** For an authenticated user who is not a device owner, the restricting of the contents of the second file system may be performed before beginning the process of authenticating the user. The restricting of the contents of the second file system may be performed after determining that the user is not a device owner.

**[0011]** The determining of whether the user is a device owner includes ascertaining ownership of a root directory of the first file system. The determining of whether the user is a device owner may include referencing access control rights stored in a non-volatile memory, with the non-volatile memory being either within or outside of the first file system.

**[0012]** The first file system may be an NTFS, and the second file system may be an NTFS. The first file system may be an ext2 file system, and the second file system may be an ext2 file system. The first file system may be an NTFS, and the second file system may appear as a FAT file system.

**[0013]** In accordance with another example embodiment, a storage device has a first memory module, a second memory module, an authentication module, and a controller. The first memory module is operative to store a first file system, the first file system being a native file system of the storage device. The second memory module is operative to store generated file system structures. The authentication module is operative to determine an identity of a user. The controller is operative to activate the authentication module and to provide to the user either the first file system or a second file system, depending on the identity of the user as determined by the authentication module. The providing of the second file system includes generating data based on file system structures of the first file system, according to the identity of the user and access control rights of the user. The generated data is stored in the second memory module.

**[0014]** In this embodiment, the first file system of the storage device may include an algorithm and a supporting for determining access control rights. The second file system may be of a type that does not include an algorithm or a supporting data structure for determining access control rights.

**[0015]** The first file system may reside in NAND flash, NOR flash, or another type of memory. The second memory module may reside in volatile memory, NAND flash, NOR flash, or another type of memory. The authentication module may be constituted by hardware, software, firmware or any combination thereof, in any manner known to those of skill in the art. The controller may be operative to provide the second file system in the same format as that of the first file system.

**[0016]** The first file system of the storage device may be an NTFS, and the second file system may be an NTFS. The first file system may be an ext2 file system, and the second file system may be an ext2 file system. The first file system may be an NTFS, and the second file system may appear as a FAT file system.

**[0017]** In accordance with yet another example embodiment, a controller for a storage device includes a first interface, a second interface, and logic. The first interface is for communication with a host. The second interface is for communication with a first memory module, an authentication module, and a second memory module, the first memory module being operative to store a first file system, which is a native file system of the storage device, the authentication module being operative to determine the identity of a user, and the second memory module being operative to store data. The logic is operative to activate the authentication module and to provide to the user either the first file system or a second file system, depending on the identity of the user as determined by the authentication module. The providing of the second file system includes generating data based on file system structures of the first file system, according to the identity of the user and access control rights of the user. The data is stored in a second memory module.

**[0018]** The first and second interfaces of the controller may be implemented in hardware, software, firmware or any combination thereof, in any manner known to those of skill in the art. The first file system of the controller may include an algorithm and supporting data structure for determining access control rights. The second file system may be of a type that does not include an algorithm or a supporting data structure for determining access control rights.

**[0019]** The logic may be operative to provide the second system in the same format as that of the first file system. The first file system of the controller may be an NTFS, and the second file system may be an NTFS. The first file system may be an ext2 file system, and the second file system may be an ext2 file system. Alternatively, the first file system may be an NTFS, and the second file system may appear as a FAT file system.

**[0020]** These and other embodiments, features, aspects and advantages of the present invention will become better understood from the description herein, appended claims, and accompanying drawings as hereafter described.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0021]** The accompanying drawings, which are incorporated in and constitute a part of this specification illustrate various aspects of the invention and together with

the description, serve to explain its principles. Wherever convenient, the same reference numbers will be used throughout the drawings to refer to the same or like elements. The accompanying drawings are briefly described below, wherein:

Fig. 1A illustrates a flow chart outlining a method of providing a file system in a storage device in accordance with a first example embodiment;

Fig. 1B illustrates a flow chart outlining an alternate method of providing a file system in a storage device in accordance with a second example embodiment;

Fig. 2 illustrates a flow chart outlining the process of providing a second file system to a host in response to a host's read request when;

Fig. 3 illustrates a storage device in accordance with another example embodiment; and

Fig. 4 illustrates a controller that may be implemented in the storage device of Fig. 3.

## DETAILED DESCRIPTION

**[0022]** The following description is provided in the context of this Application for Letters Patent and its requirements to enable a person of ordinary skill in the art to make and use the claimed invention. Various modifications to and equivalents of the embodiments described and shown are possible and various generic principles defined herein may be applied to these and other embodiments. Thus, the claimed invention is to be accorded the widest scope consistent with the principles, features and teachings disclosed herein.

**[0023]** Described first are example embodiments of methods of providing file systems in storage device systems based on the access rights of the user. Described next is a method processing a host's read request when a second file system is presented. Also described is a storage device is described, which provides file systems based on access rights of a user. Then, a controller is described that may be implemented in the storage device.

**[0024]** In the embodiments described herein, a "file system" may be an implementation of a methodology for storing and organizing computer files. A file system may include a set of abstract data types and metadata that are implemented for the storage, hierarchical organization, manipulation, navigation, access, and retrieval of data. The abstract data types and metadata form a "directory tree" through which the computer files can be accessed, manipulated and launched. A "directory tree" typically includes a root directory and subdirectories. A directory tree is stored in the file system as a "directory file." The set of metadata, directory files, or any subset thereof included in a file system is called herein a "file system structure." Such file system, therefore, includes data files and a file system structure that facilitate accessing, manipulating and launching the data files.

**[0025]** Fig. 1A illustrates an example embodiment of a method of providing a file system in a storage device. Fig. 1B illustrates an alternate example embodiment of such method. In both embodiments, a file system is provided to a user based on the user's rights.

**[0026]** One example scenario invoking these methods occurs when a user connects to the host a portable storage device, such as a USB flash drive (UFD), a Secure Digital (SD) card, a MultiMedia Card (MMC), a miniSD or a MicroSD. "USB" stands for Universal Serial Bus, which is an external peripheral interface standard for communicating between a computer and external peripherals over cables using bi-serial transmission. "SD" stands for Secure Digital™. MMC, SD, miniSD, and microSD are exemplary flash storage devices that are used with a variety of host devices such as multimedia players (e.g., MP3 and MP4 players), digital cameras, computer laptops, Global Positioning System ("GPS") devices, and so on.

**[0027]** Various storage devices can be adapted to begin such methods automatically upon such connection, and examples of such devices are later described below. Data stored in such a storage device is represented or organized according to a native file system" (also referred to herein as a "first file system"), which is a representation or organization of the data that is physically stored on the storage device. Under circumstances where the user has rights to only part of the data in the storage device, the data stored in the storage device is also represented or organized in what will be referred to as a "second file system," which term as used in this application refers to a representation or organization of the data based on the limited rights of the user. For example, the second file system may not contain, or may contain but not present to the user, all of the data files contained/presented by the first or native file system, as explained below.

**[0028]** The first step in the method of Fig. 1A and the second step in the method of Fig. 1B are authentication steps. "Authentication" is the process of determining the identity of an individual. For example, a user may be authenticated based on a username and a password. Alternatively, biometric hardware may be implemented to provide authentication. (This process is to be distinguished from the process of "authorization," which is the process of giving an authenticated user access to system objects based on the user's identity.) Some users cannot be authenticated, but the methods of Figs. 1A and 1B have provisions nonetheless for providing file systems to such users when they attempt to access contents of storage devices. To this end, both methods include the step of beginning a process of authenticating a user. (Step S1.) (The step preceding Step S1 in Fig. 1B will be discussed later.)

**[0029]** After initiating the authentication process of

Step S1, a determination is made as to whether the authentication process was successful. (Step S2.) An example situation in which authentication does not succeed is one where a user is not recognized. This can happen if the username, password, or both are not recognized or do not match any expected username /password combination. Another example situation in which authentication does not succeed is simply when the authentication process terminates prematurely, for whatever reason (e.g. software failure on the host).

**[0030]** If the authentication process is not successful, sector access criteria (i.e. identifying which sectors the user will be given access to) are established based on public access rights. (Step S3.) Accordingly, the user may not have access to all files in the native or first file system, i.e. the user may not have access to all files stored on the storage device. Access rights (or limitations thereof) may be determined by algorithms, which reference supporting data structures, for example, access control lists, directory trees, external databases, or any combination of such resources. The native file systems themselves may have the algorithms for determining access control rights. For example, if the first file system of the storage device is in NTFS 3.0 or higher format, the security_ID field within the master file table record (MFT) and corresponding security file ($Secure) entry are used as inputs into the access control algorithm defined within NTFS. According to such algorithm, access may be granted only to sectors allocated to files for which the security_ID field of the corresponding MFT record indicates an Access Control List (ACL) that has an access control entry (ACE) in which the Security ID (SID) is S-1-1-0 (Everyone) or S-1-5-7 (Anonymous user), and the ACE allows access, and for which no other ACE denies access. The storage device stores the sector access criteria (Step S4), for example, in its RAM, for use in providing a second file system in response to a sector read request from a host, as discussed below. The second file system would show as available to the user (who was not successfully authenticated) only the files which users who are defined as "anonymous users" are allowed to access. The methods of Figs. 1A and 1B may end at this point.

**[0031]** If the authentication of Step S2 is successful, the user is deemed to be an authenticated user, and a determination is made as to whether the user is a device owner. (Step S5.) The device owner may be the owner or primary user of a personal storage device or a member of the Administrators group of a corporate-controlled personal storage device. The present methods accommodate both situations when the device owner is an individual user and situations when the "owner" is any one of a group of users. The device owner status may be determined by ascertaining the ownership of the root directory of a first file system. Alternately, the device owner status may be determined by reading owner information from access control rights stored in a non-volatile memory. The access control rights may be specified as rules outside the file system or as rights inside of or outside of the file system. If the user is a device owner, the first file system is provided to the user (Step S6), and the methods end at this point.

**[0032]** In the method of Fig. 1A, if the authenticated user is not a device owner, sector access criteria are established based on the user's identity determined during the authentication process. (Step S7.) For example, if the native file system of the storage device is in NTFS format, access may be granted to sectors in which the security_id fields of the corresponding MFT records have values that indicate that the particular user has access to those sectors. The sector access criteria may be based on a single rule that applies to an entire class of users or on a set of rules that apply respectively to different subgroups of users or to different individual users. The process flow then proceeds to Step S4, in which the storage device stores the sector access criteria for use when processing a sector read request. A second file system is now provided to the user, and the method ends at this point. The second file system would show as available to the user (who was successfully authenticated but is not a device owner) only the files which the particular user is allowed to access.

**[0033]** In the method of Fig. 1B, the storage device establishes sector access criteria for a variety of types of users and/or particular users who can be authenticated but are not device owners (Step S9), and this step is executed before beginning the authentication process in Step S1. The sector access criteria may be established when the native file system is created or updated by the device owner. If later an authenticated user is determined in Step S5 not to be a device owner, the sector access criteria for the particular user are selected from the criteria established in Step S9. (Step S10.) The process flow then leads to Step S4, in which the storage device stores the sector access criteria for use when processing a sector read request. A second file system is now provided to the user, and the method can end at this point. The second file system would show as available to the user (who was not successfully authenticated) only the files which the user (the public) is allowed to access.

**[0034]** After a second file system is provided, a host's read request for a sector in the memory of a storage device implementing the present invention may be processed according to the following example procedure described with reference to Fig. 2. The process begins when the storage device receives a sector read request from the host. (Step S1.) In response, the storage device generates a copy of the directory entry that contains the requested sector from the native system. (Step S2.) The copied sector may be stored temporarily in RAM.

**[0035]** From the copied directory entry, the storage device determines whether access to the requested sector is permitted to the particular user according to the established sector access criteria. (Step S3.) For example, if the native file system is in NTFS format, the directory entry is an MFT record, and the storage device checks

the established sector access criteria to determine whether the security_id field within the MFT record has a value that permits the user to have access to the sector. If access to the sector is permitted, the storage device returns the sector to the host (Step S4), and the process ends.

**[0036]** If instead the storage device determines in Step S3 that access to the sector is not permitted to the particular user, the storage device modifies data, i.e., the sector in the copy of the directory entry, to prevent the host from retrieving the unmodified sector from the native file system. (Step S5.) For example, in NTFS, the storage device could change attributes by removing standard information or filename attributes or by changing the magic number (a code uniquely identifying the type of record) of the directory entry. With attributes changed accordingly, the file no longer appears in directory listings and cannot be accessed using standard calls to file system application program interface (API), nor does the file appear to be valid when forensic software tools are used to access the storage device. That is, the file is hidden from the user. The process then flows to Step S4, where the storage device returns the modified sector to the host, at which point the process can end.

**[0037]** Various options for cache management are available as known to those skilled in the art. For example, the modified sectors may be stored in the RAM of the storage device to save the resources that would have been required to modify a sector (Step S6) again after the first time a host requests it. This conservation of resources is possible, if no intervening write operation necessitating a re-calculation of the modification occurs between the first and subsequent host requests.

**[0038]** Although the process of Fig. 2 can be applied to a storage device that provides both the first (native) and the second file systems in the same format, for example, in NTFS format (as discussed above), the application of this process is not limited to representing both file systems in the same format or in NTFS format.

**[0039]** For example, the first (native) and second file systems may both be in ext2 format. In this case, the directory entry referenced and/or processed in the various steps is an inode. In Step S3, the determination of whether access to the inode is permitted to a particular user is performed by the storage device checking the owner information within the inode.

**[0040]** Then again, in the case in which the first and second file systems do not have the same format, the first file system can be in NTFS and the second file system can appear as a FAT file system by generating structures such as file allocation tables, a boot parameter block and directories for the second file system. Because FAT file systems do not have algorithms and supporting data structures for determining access control rights, the second file system of this implementation does not include such an algorithm and supporting data structure.

**[0041]** In the case in which the first and second file systems have different formats, all the file system structures for the second file system are generated before the storage device is made accessible for read by the host. However, where the first and second file systems have the same formats, it is not the case that all the file system structures need to be generated before the storage device is made accessible for read by the host; rather, a portion or all of the file system structures may be generated in response to read requests from the host.

**[0042]** Another example embodiment of the present invention is a storage device 10 as shown in Fig. 3. The storage device 10 may be implemented, for example, as a UFD, a SD card, a MMC, a miniSD, or a MicroSD. The storage device 10 includes a first file system 12, which is a native file system, stored in a first memory module 14. All files in the first file system 12 are accessible by a device owner, and a subset of those files is accessible also to authenticated users who are not device owners, and in this embodiment a different subset of those files, as determined by public access rights, is accessible to users who are not authenticated. In some alternate embodiments the same set of files may be accessible to both (1) authenticated users who are not device owners and (2) unauthenticated users.

**[0043]** For authenticating users, the storage device 10 includes an authentication module 16, which determines the identity of a user by reference to a user database 18. The authentication module 16 may be embodied as containing hardware, software, firmware, or a combination thereof, which determines authenticity based on identifying information provided by a user.

**[0044]** For example, the authentication module 16 may be embodied to include software code executable by controller 20. The software code compares a username and password received as input through an external interface of the storage device 10 with usernames and corresponding passwords stored in the user database 18. If the software code finds the input username and password in the user database 18, the software code responds with the user's identity. Otherwise, the software code indicates that the user is not authenticated.

**[0045]** Alternatively, the authentication module 16 may be embodied as biometric hardware. For example, the hardware may include a fingerprint scanner or a voice recognition sensor that receives user input in the form of a finger print or audio signal, respectively. The hardware compares the user input to user data in the user database 18 and responds with the user's identity when such information is available.

**[0046]** The storage device 10 has a controller 20, which is operative to activate the authentication module 16 to identify a user and then determine whether to provide to the user either the first file system 12 or a second file system, depending on the user's identity. The controller 20 is operative to perform these tasks, because it has access to program code residing within a ROM mask 22 internal to controller 20, as shown in Fig. 4. The program code residing within the ROM mask 22 may be embodied to direct the controller 20 to operate as described earlier

with respect to methods represented by Figs. 1A, 1B, and 2.

**[0047]** A processor 24 also internal to controller 20 receives host read requests through an interface 26, processes the requests according to the logic within the ROM mask 22, and accesses the other elements of the storage device 10 accordingly though another interface 30 of the controller 20. As shown in Figs. 3 and 4, the interface 26 of the controller 20 communicates directly with an interface 28 of the storage device 10, which is an interface to the host. Example interfaces available as interface 28 are those that comply with the USB, SD card, MMC, miniSD, or MicroSD standards. Note that, although in this embodiment the logic resides as firmware in a ROM mask 22, the logic may reside elsewhere, such as in a separate ASIC (as hardware). The logic may also be implemented as firmware, for example, as flash-based code running on a general purpose core.

**[0048]** The storage device 10 has a second memory module, a RAM 32, which may be for example a Double Data Rate (DDR) RAM. RAM 32 stores data, i.e., directory entries 33 derived from or based on directory entries of the native file system, which are used for providing second file systems according to the identity and access control rights of the user. Example directory entries may be those described earlier with respect to the methods represented by Figs. 1A, 1B, and 2. The access control rights may be specified for example by an access control list (ACL) 34 residing in for example a third memory module 36. In alternative embodiments, the first file system 12 may include an algorithm and supporting data structure for determining access control rights.

**[0049]** Also, although in the present embodiment the second memory module 32 is a volatile memory for storing generated file system structures it is not limited to this structure. The second memory module could even be provided instead within a hard drive of the host. The second memory module 32 in the present embodiment is directly addressable by the controller 20. In alternate embodiments, in which the second memory module is not directly addressable by the controller 20, a memory management unit (MMU) or equivalent interface module is used to effect the addressing. Other types of memory that may be used as the second memory module include Synchronous Dynamic Random Access Memory (SDRAM) and DDR Memory.

**[0050]** The present description refers to the memory modules of the first file system 12, the RAM 32, and the access control list 34 as the first, second, and third memory modules. There may also be a further memory module 38 for holding the user database 18. Despite the designation of these four memory modules as separate memory modules in the present discussion, any or all of these memory modules may be implemented together as a single module. For example, the memory modules 14, 36, and 38 may be implemented as elements of a single flash memory unit as indicated by the broken-lined box in Fig. 3. Having thus described exemplary embodiments, it will be apparent that various alterations, modifications, and improvements will readily occur to those skilled in the art. For example, access control rights may be determined from an owner/group/world specification like that found in Linux file systems, in which a set of access rights is assigned to the owner, another set of access rights is assigned to any user in the same group as the owner, and a third set of rights is assigned to any user ("world") who is not a member of that group or who is not authenticated. A file accessible to all users is considered "world-accessible."

In sum, although various embodiments of the present invention have been described in considerable detail alternations, modifications, and improvements of the disclosed embodiments, though not expressly described above, are nonetheless intended and implied to be within the scope of the claims.

Accordingly, the foregoing discussion is intended to be illustrative only and the invention is limited and defined only by the following claims and equivalents thereto.

## Claims

**1.** A method of providing a file system, the method comprising:

in a data storage device (10) operatively coupled to a host device, performing:

commencing authentication of a user to the data storage device (10);

if the authentication does not succeed, providing to the host device coupled to the data storage device (10) a third file system, contents of the third file system being restricted to files authorized by public access rights; and

if the authentication does succeed, determining whether the user is authenticated as having full rights to the data stored at the storage device by ascertaining ownership of a root directory of a first file system (12), wherein:

in response to the user being authenticated as having full rights, providing to the host device the first file system (12), the first file system (12) being a native file system of the data storage device (10), wherein the third file system includes a subset of the contents of the first file system; and

in response to the user not being authenticated as having full rights, providing to the host device a second file system, the second file system being restricted to files that the user is authen-

ticated as being authorized to access.

2. The method of claim 1, wherein the user being authenticated as having full rights includes referencing access control rights stored in a non-volatile memory (36).

3. The method of claim 1, wherein the first file system is a new technology file system, hereinafter referred to as NTFS, and the second file system is an NTFS.

4. The method of claim 1, wherein the first file system is a second extended file system, hereinafter referred to as ext2 file system, and the second file system is an ext2 file system.

5. The method of claim 1, wherein the first file system is a new technology file system and the second file system appears as a file allocation table file system.

6. The method of claim 1, wherein the contents of the second file system include a reference to at least one file.

7. A storage device (10), the storage device (10) comprising:

a first memory module (14) operative to store a first file system (12), the first file system (12) being a native file system of the storage device (10);
a second memory module (32) operative to store generated data;
an authentication module (16) operative to authenticate a user; and
a controller (20) operative to activate the authentication module (16) and in response to the user not being authenticated providing to the host device coupled to the data storage device (10) a third file system, contents of the third file system being restricted to files authorized by public access rights; and
in response to the user being authenticated determining whether the user is authenticated as having full rights to the data stored at the storage device by ascertaining ownership of a root directory of a first file system, wherein:

in response to the user being authenticated as having full rights, providing to the host device a first file system (12), the first file system (12) being a native file system of the data storage device (10), wherein the third file system includes a subset of the contents of the first file system; and
in response to the user not being authenticated as having full rights, providing to the host device a second file system, the second file system being restricted to files that the user is authenticated as being authorized to access.

**Patentansprüche**

1. Verfahren zur Bereitstellung eines Dateisystems, wobei das Verfahren Folgendes umfasst:

in einer Datenspeichervorrichtung (10), operativ an eine Host-Vorrichtung gekoppelt, die Durchführung von Folgendem:

Beginnen der Authentifizierung eines Benutzers gegenüber der Datenspeichervorrichtung (10);
wenn die Authentifizierung nicht erfolgreich ist, Bereitstellen für die Host-Vorrichtung, die an die Datenspeichervorrichtung (10) gekoppelt ist, eines dritten Dateisystems, wobei Inhalte des dritten Dateisystems auf Dateien beschränkt sind, die durch öffentliche Zugriffsrechte autorisiert werden; und
wenn die Authentifizierung erfolgreich ist, Bestimmen, ob der Benutzer so authentifiziert wurde, dass er volle Rechte für die in der Speichervorrichtung gespeicherten Daten hat, indem der Besitz eines Stammverzeichnisses eines ersten Dateisystems (12) sichergestellt wird, wobei:

als Reaktion darauf, dass der Benutzer so authentifiziert wurde, dass er volle Rechte hat, der Host-Vorrichtung das erste Dateisystem (12) bereitgestellt wird, wobei das erste Dateisystem (12) ein natives Dateisystem der Datenspeichervorrichtung (10) ist, wobei das dritte Dateisystem eine Teilmenge der Inhalte des ersten Dateisystems einschließt; und
als Reaktion darauf, dass der Benutzer nicht so authentifiziert wurde, dass er volle Rechte hat, der Host-Vorrichtung ein zweites Dateisystem bereitgestellt wird, wobei das zweite Dateisystem auf Dateien beschränkt ist, für die der Benutzer so authentifiziert wurde, dass er autorisiert ist, um auf sie zuzugreifen.

2. Verfahren nach Anspruch 1, wobei die Authentifizierung des Benutzers, so dass er volle Rechte hat, die Referenzierung von Zugriffsteuerungsrechten einschließt, die in einem nicht flüchtigen Speicher (36) gespeichert sind.

3. Verfahren nach Anspruch 1, wobei das erste Datei-

system ein New Technology File System ist, nachfolgend als NTFS bezeichnet, und das zweite Dateisystem ein NTFS ist.

**4.** Verfahren nach Anspruch 1, wobei das erste Dateisystem ein Second Extended File System ist, nachfolgend als ext2 File System bezeichnet, und das zweite Dateisystem ein ext2 File System ist.

**5.** Verfahren nach Anspruch 1, wobei das erste Dateisystem ein New Technology File System ist und das zweite Dateisystem als ein File-Allocation-Table-Dateisystem auftritt.

**6.** Verfahren nach Anspruch 1, wobei die Inhalte des zweiten Dateisystems einen Verweis auf mindestens eine Datei einschließen.

**7.** Speichervorrichtung (10), wobei die Speichervorrichtung (10) Folgendes umfasst:

ein erstes Speichermodul (14), das operativ ist, um ein erstes Dateisystem (12) zu speichern, wobei das erste Dateisystem (12) ein natives Dateisystem der Speichervorrichtung (10) ist;
ein zweites Speichermodul (32), das operativ ist, um generierte Daten zu speichern;
ein Authentifizierungsmodul (16), das operativ ist, um einen Benutzer zu authentifizieren; und
eine Steuereinheit (20), die operativ ist, um das Authentifizierungsmodul (16) zu aktivieren und als Reaktion darauf, dass der Benutzer nicht authentifiziert wurde, der Host-Vorrichtung, die an die Datenspeichervorrichtung (10) gekoppelt ist, ein drittes Dateisystem bereitzustellen, wobei Inhalte des dritten Dateisystems auf Dateien beschränkt sind, die durch öffentliche Zugriffsrechte autorisiert werden; und
als Reaktion darauf, dass der Benutzer authentifiziert wurde, zu bestimmen, ob der Benutzer so authentifiziert wurde, dass er volle Rechte für die in der Datenspeichervorrichtung gespeicherten Daten hat, indem der Besitz eines Stammverzeichnisses eines ersten Dateisystems sichergestellt wird, wobei:

als Reaktion darauf, dass der Benutzer so authentifiziert wurde, dass er volle Rechte hat, der Host-Vorrichtung ein erstes Dateisystem (12) bereitgestellt wird, wobei das erste Dateisystem (12) ein natives Dateisystem der Datenspeichervorrichtung (10) ist, wobei das dritte Dateisystem eine Teilmenge der Inhalte des ersten Dateisystems einschließt; und
als Reaktion darauf, dass der Benutzer nicht so authentifiziert wurde, dass er volle Rechte hat, der Host-Vorrichtung ein zweites Dateisystem bereitgestellt wird, wobei das zweite Dateisystem auf Dateien beschränkt ist, für die der Benutzer so authentifiziert wurde, dass er autorisiert ist, um auf sie zuzugreifen.

**Revendications**

**1.** Une méthode de réalisation d'un système de fichier, la méthode comprenant :

dans un dispositif de stockage des données (10) accouplé de façon opérationnelle avec un dispositif hôte, l'exécution des étapes suivantes :

commencer l'authentification d'un utilisateur pour le dispositif de stockage des données (10) ;
si l'authentification échoue, fournir au dispositif hôte accouplé avec le dispositif de stockage des données (10) un troisième système de fichier, le contenu du troisième système de fichier étant réservé pour des fichiers autorisés par des doits d'accès publics ; et
si l'authentification réussit, déterminer si l'utilisateur est authentifié comme jouissant de droits intégraux pour les données stockées au dispositif de stockage, en établissant le titre à un répertoire racine d'un premier système de fichier (12), dans lequel :

en réponse à l'authentification de l'utilisateur comme jouissant de droits intégraux, fournir au dispositif hôte le premier système de fichier (12), le premier système de fichier (12) étant un système de fichiers natif du dispositif de stockage des données (10), le troisième système de fichier comprenant un sous-ensemble du contenu du premier système de fichier ; et
en réponse au fait que l'utilisateur n'est pas authentifié comme jouissant de droits intégraux, fournir au dispositif hôte un deuxième système de fichier, le deuxième système de fichier étant réservé pour des fichiers auxquels l'utilisateur a été authentifié comme étant autorisé à y accéder.

**2.** La méthode selon la revendication 1, l'authentification de l'utilisateur comme jouissant de droits intégraux comprenant des droits de contrôle de l'accès à des références placés dans une mémoire rémanente (36).

3. La méthode selon la revendication 1, le premier système de fichier étant un système de fichiers d'une technologie nouvelle, désignée ci-après NTFS, et le deuxième système de fichiers étant un NTFS.

4. La méthode selon la revendication 1, le premier système de fichier étant un deuxième système de fichiers étendus, désigné ci-après système de fichier ext2, et le deuxième système de fichiers est un système de fichier ext2.

5. La méthode selon la revendication 1, le premier système de fichier étant un système de fichiers à technologie nouvelle, et le deuxième système de fichier s'affichant comme un système de fichiers à table d'attribution de fichiers.

6. La méthode selon la revendication 1, le contenu du deuxième système de fichier comprenant une référence à au moins un fichier.

7. Un dispositif de stockage (10), le dispositif de stockage (10) comprenant :

un premier module de mémoire (14) assurant le stockage d'un premier système de fichier (12), le premier système de fichier (12) étant un système de fichier natif du dispositif de stockage (10) ;
un deuxième module de mémoire (32) assurant le stockage de données produites ;
un module d'authentification (16) servant à authentifier un utilisateur ; et
un contrôleur (20) assurant l'activation du module d'authentification (16), et, en réponse au fait que l'utilisateur n'a pas été authentifié, fournissant au dispositif hôte accouplé avec le dispositif de stockage (10) un troisième système de fichier, le contenu du troisième système de fichier étant limité à des fichiers autorisés par des droits d'accès publics ; et
en réponse à l'authentification de l'utilisateur, déterminant si l'utilisateur a été authentifié comme jouissant de droits intégraux aux données stockées dans le dispositif de stockage, en établissant le titre à un répertoire racine d'un premier système de fichier, dans lequel :

en réponse à l'authentification de l'utilisateur comme jouissant de droits intégraux, un premier système de fichier (12) est fourni au dispositif hôte, le premier système de fichier (12) étant un système de fichier natif du dispositif de stockage (10) de données, le troisième système de fichier comprenant un sous-ensemble du contenu du premier système de fichier ; et
en réponse au fait que l'utilisateur n'est pas authentifié comme jouissant de droits intégraux, un deuxième système de fichier est fourni au dispositif hôte, le deuxième système de fichier étant limité à des fichiers auxquels l'utilisateur a été authentifié comme étant autorisé à y accéder.

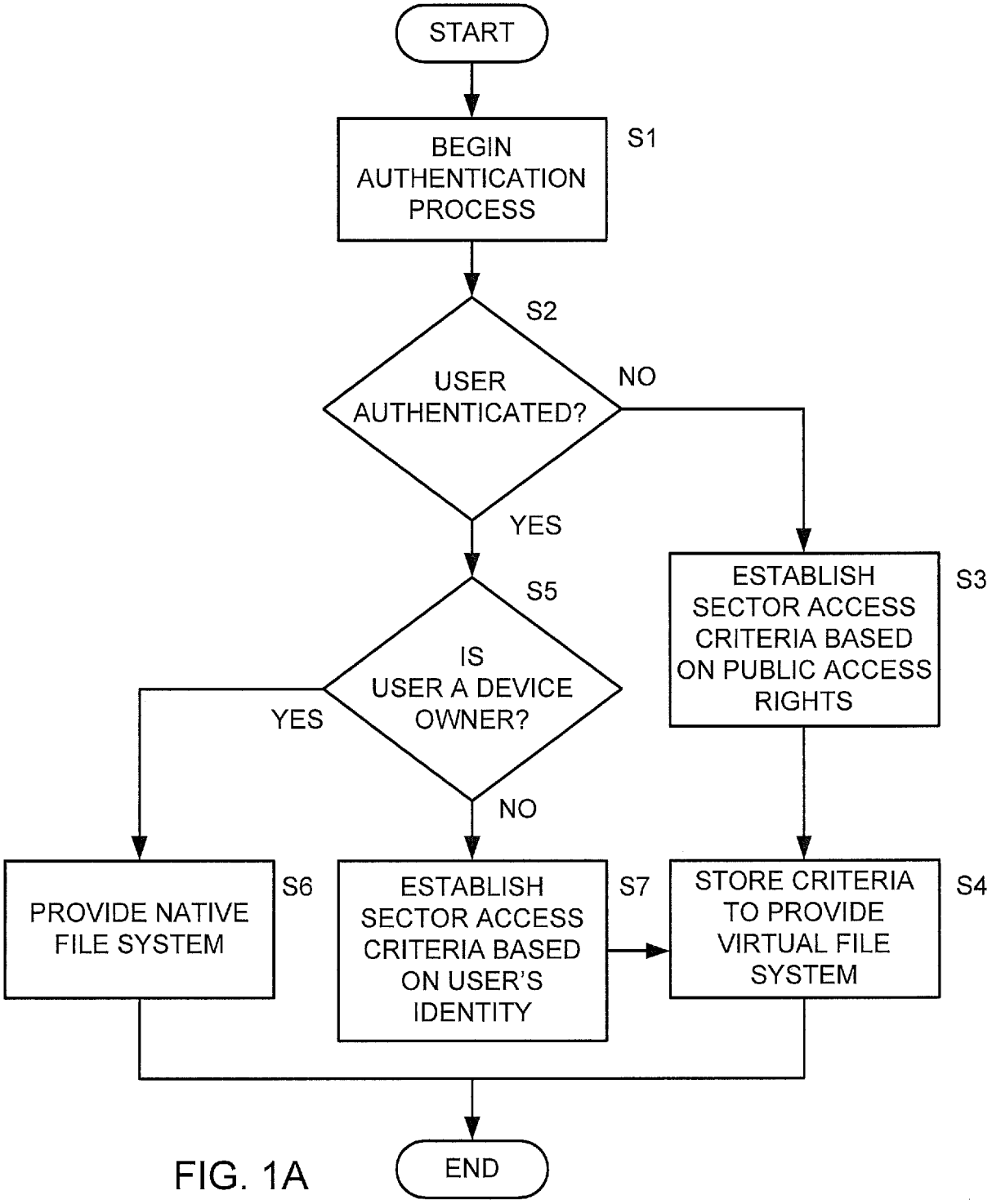
START
BEGIN AUTHENTICATION PROCESS
S1
S2
USER AUTHENTICATED?
NO
YES
S5
IS USER A DEVICE OWNER?
YES
NO
ESTABLISH SECTOR ACCESS CRITERIA BASED ON PUBLIC ACCESS RIGHTS
S3
PROVIDE NATIVE FILE SYSTEM
S6
ESTABLISH SECTOR ACCESS CRITERIA BASED ON USER'S IDENTITY
S7
STORE CRITERIA TO PROVIDE VIRTUAL FILE SYSTEM
S4
END

FIG. 1A

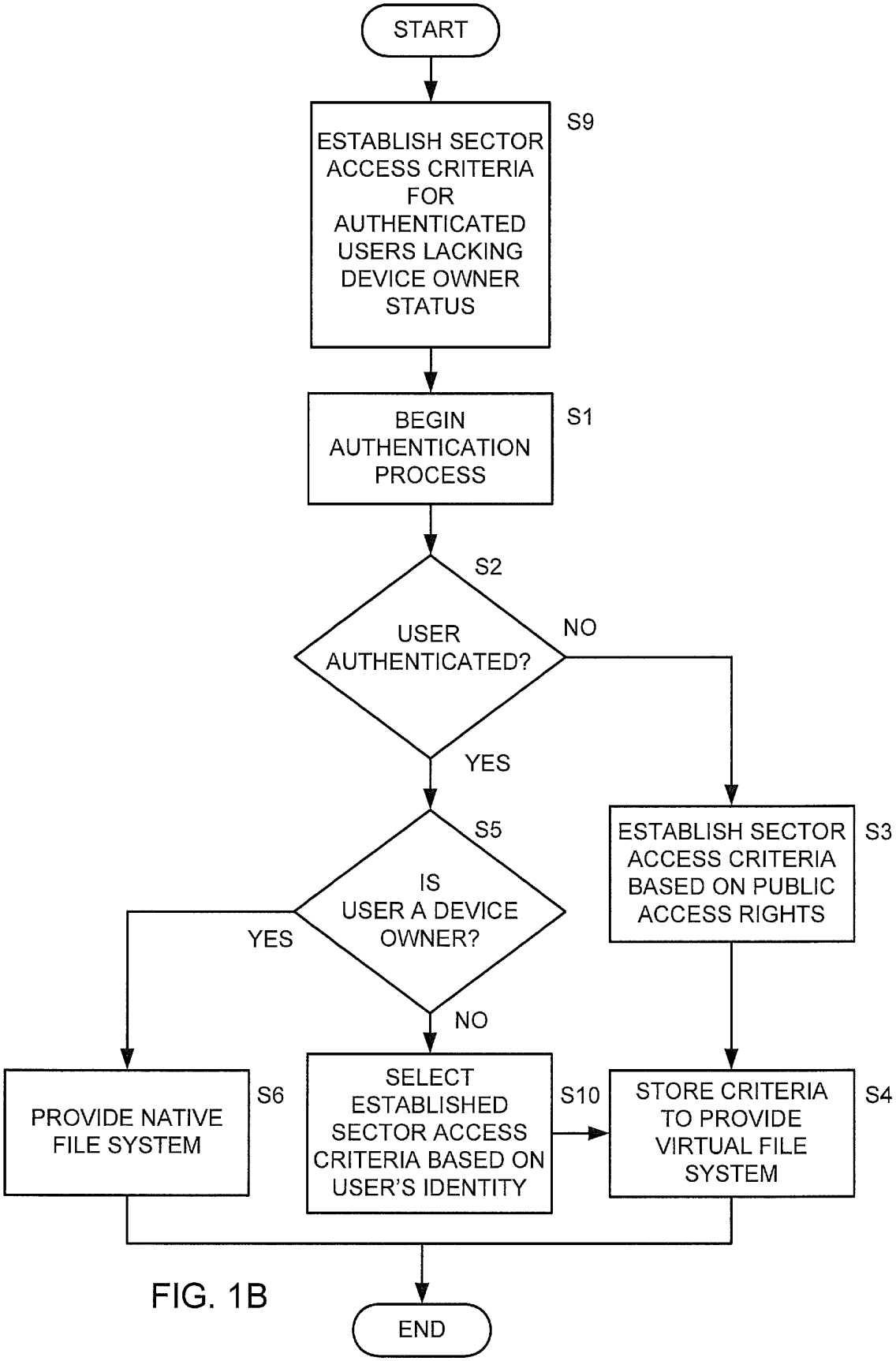
START
ESTABLISH SECTOR ACCESS CRITERIA FOR AUTHENTICATED USERS LACKING DEVICE OWNER STATUS
S9
BEGIN AUTHENTICATION PROCESS
S1
S2
USER AUTHENTICATED?
NO
YES
S5
IS USER A DEVICE OWNER?
YES
NO
ESTABLISH SECTOR ACCESS CRITERIA BASED ON PUBLIC ACCESS RIGHTS
S3
PROVIDE NATIVE FILE SYSTEM
S6
SELECT ESTABLISHED SECTOR ACCESS CRITERIA BASED ON USER'S IDENTITY
S10
STORE CRITERIA TO PROVIDE VIRTUAL FILE SYSTEM
S4
END

FIG. 1B

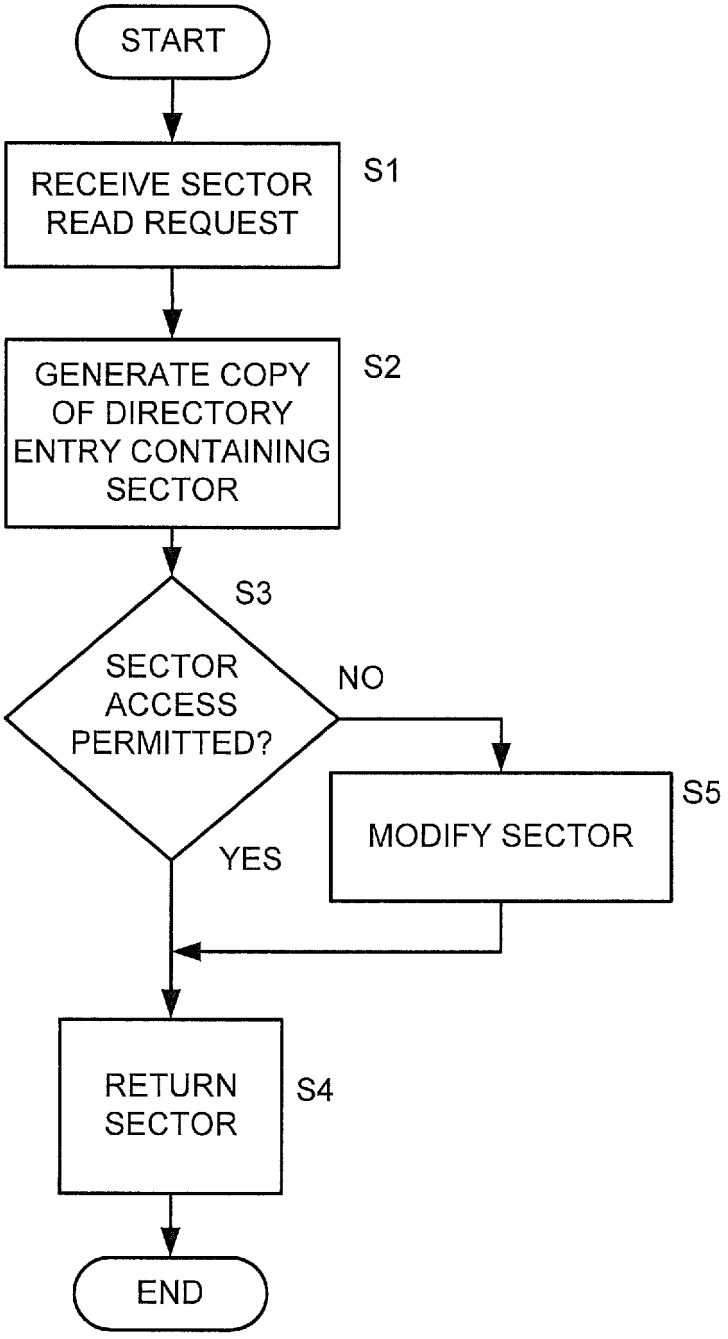
START
RECEIVE SECTOR READ REQUEST
S1
GENERATE COPY OF DIRECTORY ENTRY CONTAINING SECTOR
S2
S3
SECTOR ACCESS PERMITTED?
NO
YES
MODIFY SECTOR
S5
RETURN SECTOR
S4
END

FIG. 2

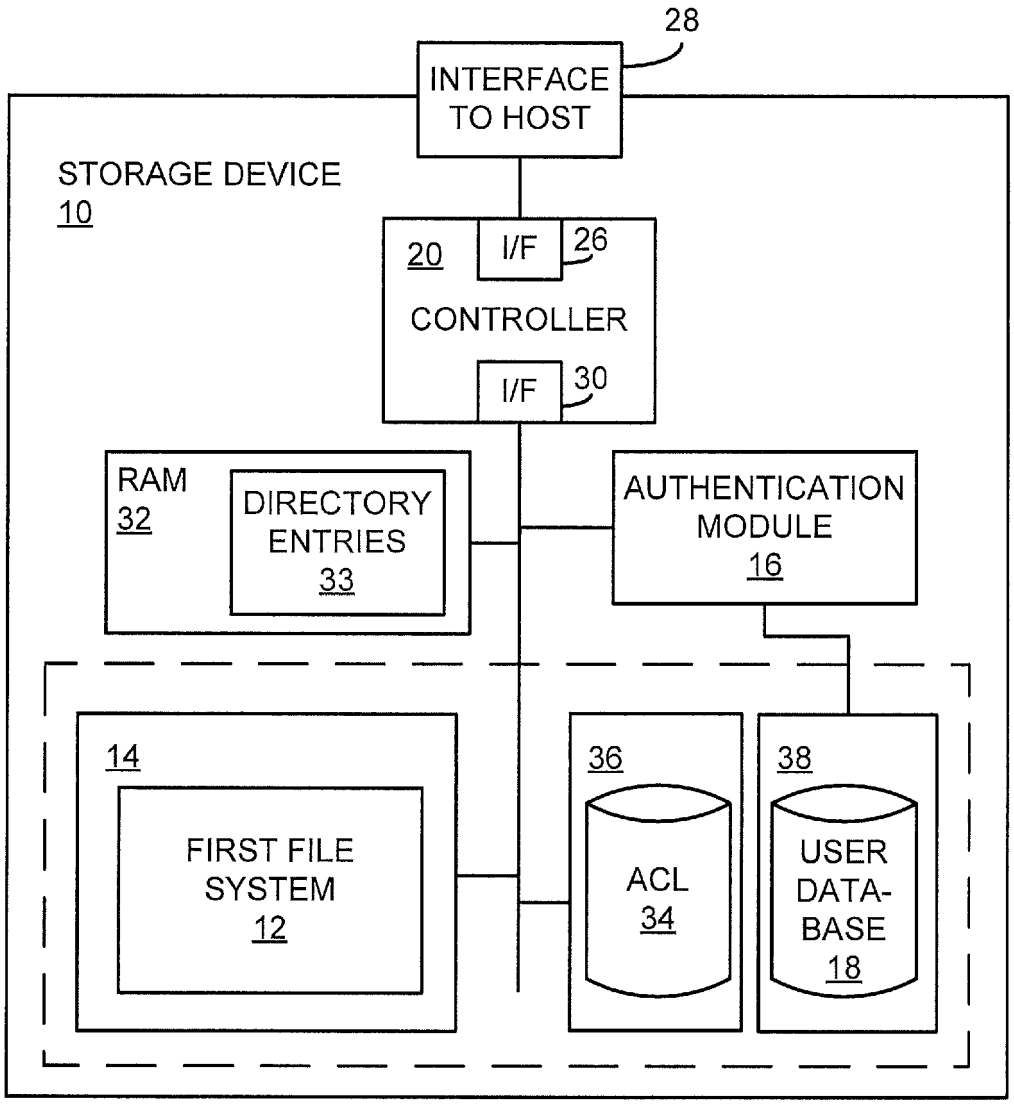
28
INTERFACE TO HOST
STORAGE DEVICE
10
20
I/F
26
CONTROLLER
I/F
30
RAM
32
DIRECTORY ENTRIES
33
AUTHENTICATION MODULE
16
14
FIRST FILE SYSTEM
12
36
ACL
34
38
USER DATA-BASE
18

FIG. 3

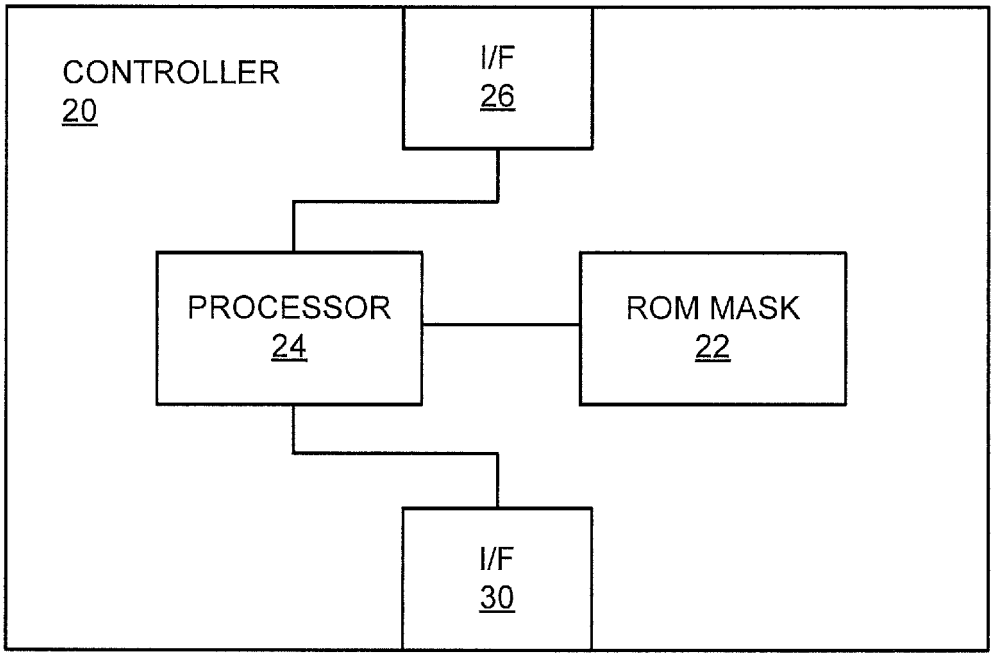


FIG. 4

## REFERENCES CITED IN THE DESCRIPTION

*This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.*

### Patent documents cited in the description


- US 20070283094 A **[0006]**